Date=14/12/2020
Lecture By= Akash Handa
Subject ⇒ React

IN PREVIOUS LECTURE **(QUICK RECAP)** **Date-10/12/2020**	**In Today's Lecture (Overview)**
componentDidMount() method	What is Filter in React? Anxious In react React Slick Quick start

What is Filter in React?

Filtering in React is pretty much what it says it is. It's the process of looping through an array and including or excluding elements inside that array based on a condition that you provide.

The caveat here is that we're not actually filtering using React. Filter is a JavaScript function that we can perform on an array type object.

Remember, we use JavaScript to write React. None of these methods are special to React. React is just the UI library.

Example 1: Filter an Array of Strings in React

This first example is quite a common scenario. Picture the scene: you're creating a search filter component to allow your users to search a list of names.

Your users type into a text field to filter that array of names based on what they're typing into that text field. It's quite a common input element these days.

Let's start off with a hard-coded array of strings. Names that you might recognize (plus a small addition):

```
const names = ['James', 'John', 'Paul', 'Ringo', 'George'];
```

Let's assume that you want to only display names that include the letter 'J' in them.

To do that, we use the filter function. We can perform the filter function inside of our JSX in a React component like so:

App.js

```
<div>
```

```
  {names.filter(name => name.includes('J')).map(filteredName => (
    <li>
      {filteredName}
    </li>
  ))}
</div>
```

Finally, we perform another function after the filter function: the map function. This is how we can output the array elements that match the filter condition to the actual user interface inside of li tags.

Let’s see the complete React component code:

App.js

```
import React from 'react';

const names = ['James', 'John', 'Paul', 'Ringo', 'George'];

function App() {
  return (
    <div>
      {names.filter(name => name.includes('J')).map(filteredName => (
        <li>
          {filteredName}
        </li>
      ))}
    </div>
  );
}

export default App;
```

Anxious In react

Axios is a library that helps us make http requests to external resources

In our React applications we often need to retrieve data from external APIs so it can be displayed in our web pages.

One way to build this feature is to use the Javascript *Fetch* API. Fetch is quite capable of retrieving external data, but it has some limitations.
A more popular way of performing this operation is to use the *Axios* library. Axios is designed to handle http requests and responses.
It's used more often than Fetch because it has a larger set of features and it supports older browsers.
Axios deals with responses using *Promises*, so it's streamlined and easy to use in our code. Axios uses methods like get() and post() that perform http GET and POST requests for retrieving or creating resources.
We install Axios in our project in the usual way, with the npm command:

```
npm install axios
```

React Slick

React slick is a carousel component built with React. It is a react port of **slick carousel**

Quick start

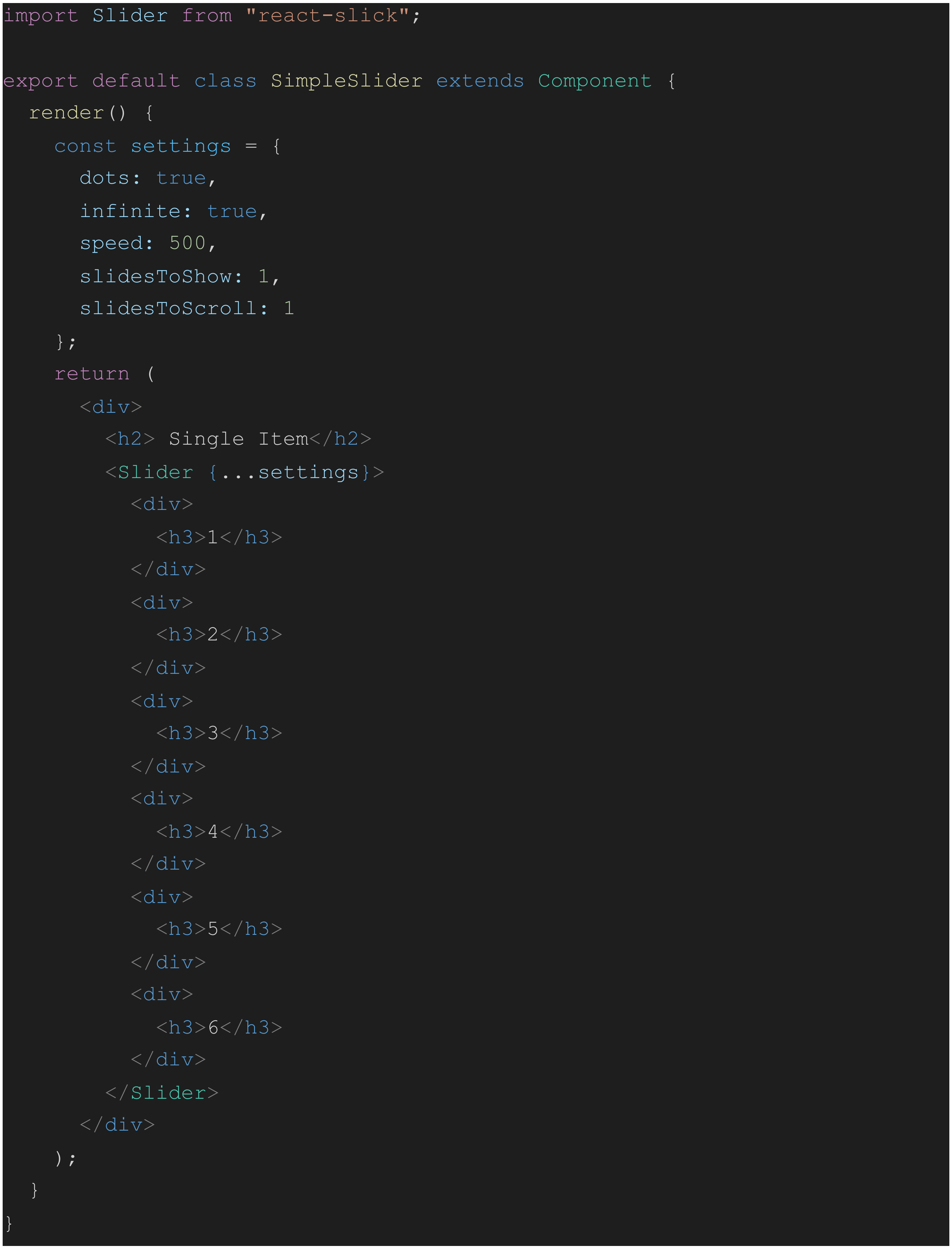
The easiest way to try out react-slick is using the **Code Sandbox example**

or you can create an index.html file and include react-slick with:

```
<link rel="stylesheet" type="text/css" charset="UTF-8"
href="https://cdnjs.cloudflare.com/ajax/libs/slick-carousel/1.6.0/slick.min.css" />
<link rel="stylesheet" type="text/css"
href="https://cdnjs.cloudflare.com/ajax/libs/slick-carousel/1.6.0/slick-theme.min.css" />
```

Complete code

```
import React, { Component } from "react";
```



```
import Slider from "react-slick";

export default class SimpleSlider extends Component {
  render() {
    const settings = {
      dots: true,
      infinite: true,
      speed: 500,
      slidesToShow: 1,
      slidesToScroll: 1
    };
    return (
      <div>
        <h2> Single Item</h2>
        <Slider {...settings}>
          <div>
            <h3>1</h3>
          </div>
          <div>
            <h3>2</h3>
          </div>
          <div>
            <h3>3</h3>
          </div>
          <div>
            <h3>4</h3>
          </div>
          <div>
            <h3>5</h3>
          </div>
          <div>
            <h3>6</h3>
          </div>
        </Slider>
      </div>
    );
  }
}
```